二維細胞神經網絡之穩定解個數研究

自小學六年級經許文化老師指導參加科展競賽後，研究數學即成了我的興趣。除了高一之外，每年做出一件研究成果參加科展，幾乎已經溶入了我的成長學習了。

此次很榮幸參加國際科展，感謝林礽堂老師的鞭策，讓我有更深一層研究的動力；感謝交通大學林松山院長和班榮超老師，提供研究題目給我，不僅提升了我的研究能力，更肯定了我的研究成果。林院長亦抽空指導我寫作方法，更讓我受益無窮。

翻譯的過程中，我承受了極大的考驗，除了台大數學研究所的卓士堯學長，鼎力幫忙外，最感謝中央研究院的葉永南教授、劉恕忠老師，及清大全任重教授的指導。

無論如何，這次科展，我是盡了全力，而且也必將是我高中生活中最難忘的記憶。

作者：戴淯琮
就讀學校：建國中學
指導老師：林礽堂　林松山

研究動機

有一篇論文研究「一維細胞神經網絡之穩定解個數」，探討直線形細胞神經為 1×3 時之變化，然後推廣到所有奇數之穩定解個數，論文中是用矩陣乘法研究。我看了這篇論文感到非常有趣，便去找老師討論，我從 L 形著手，由直線推廣到平面，我邊做邊想，也找出一些規律，於是我就把它作爲我的科展研究。

研究方向

(一)題意探索。
(二)尋找矩陣的解題。

研究過程

(一) 符號說明

爲了敘述方便，凡大寫英文字代表矩陣，小寫英文字代表實數或同時代表實數和矩陣，大寫希臘字代表一組排法，小寫希臘字代表一個排法，天干代表正負號。a_{ij} 代表矩陣 A 中第 i 列第 j 行的元，A^t 代表矩陣 A 的轉置矩陣，$a_{0j}=\sum_{i=1}^{m} a_{ij}$,$a_{i0}=\sum_{j=1}^{n} a_{ij}$,$a_{00}=\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}$,其中 A 是 $m\times n$ 矩陣，I 爲矩陣的乘法單位元素，0 爲實數或矩陣的加法單位元，$T(\Gamma,m\times n)$代表Γ排成大小爲 $m\times n$ 的圖形個數。

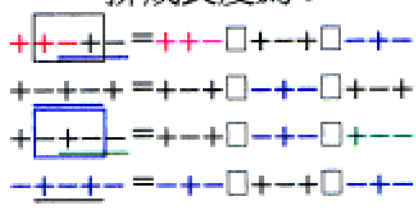
設α中最右邊的兩個正負號和β中最左邊的兩個正負號相同。不失一般性，可令$\alpha=\alpha_1$甲乙, $\beta=$甲乙β_1，則定義$\alpha\square\beta=\alpha_1 xy\beta_1$.

(二)一維 CNN 的回顧

求一維 CNN 穩定解個數的方法，可以用簡單的+–號來表示，以 1×3 爲例：將任 3 個+–號排列在直線上有，+++,++–,+–+,+––,–++,–+–,––+,–––8 種。在 8 種+–號的排法中，我們可取其中 1,2,…,8種成一組，每種排法使用的次數沒有限制，求排成不同長度時排法的個數。例如：

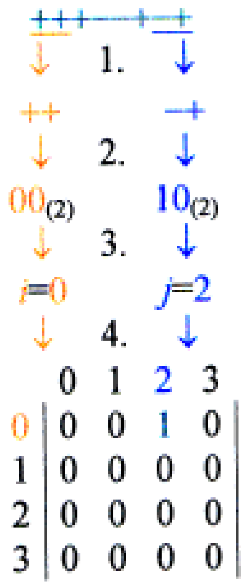
取++−,+−+,+−−,−+−為一組，排成長度為 5 時排法的個數有 4 個：

排成長度為 5



若排成長度為 6

可以發現，由長度為 5 排成長度為 6 時，最右邊的兩個+−號會影響排成的排法及個數。所以只要記錄排法最右邊的+−及個數，就可以得到延伸的排法。而當計算很長的排法時，不用把所有的+−號列出，可節省很多空間。



為了兼顧排法的記錄和相接時的演算，我們將排法記錄在一個矩陣中，記錄方法如下：

1.取出排法中最左邊和最右邊的兩個+−號，在此例中分別是 ++,−+；

2.是將其中的+用0代替,−用1代替，把這最左邊和最右邊的兩個+−號分別變成兩個 2 位數；

3.可以把這個 2 位數視為 2 進位；

4.令左邊的兩個符號經過上述步驟得到的數為 i，右邊的為 j，則第 4 步是把矩陣第 i 列第 j 行填入 1，矩陣中其他的位置填入 0，而所得的矩陣就記錄了這個排法。

如果要記錄一組排法，則可分別求出每一個的矩陣，再將那些矩陣加起來就可以了。這種記錄方式的好處在於：設 α,β 是兩個排法，γ 是 α,β 相接後的排法，A,B,C 分別代表 α,β,γ，則 $AB=C$，例如：

+++−−+−+□−++++−+−=+++−−+−++++−+−

$$\begin{pmatrix}0&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\times\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&1&0&0\\0&0&0&0\end{pmatrix}=\begin{pmatrix}0&0\times0+0\times0+1\times1+0\times0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}=\begin{pmatrix}0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}$$

由上面的性質可得出：設Γ是一組排法，A 代表Γ的矩陣，則 A^{n-2} 恰代表將Γ排成長度為 n 時的矩陣，所以只要將 A^{n-2} 中每一個數加起來，就可以知道將Γ排成長度為 n 時的總排法數。

矩陣方法中最重要精神在於只記錄直線的左右各兩個而不管中間，故不論排法有多長，都可以用一個矩陣的 16 個元來記錄。

一維矩陣方法的詳細情形可參考資料中班榮超的論文。

(三) 二維排法的矩陣解法

而 L 形只是把 1×3 彎成 L 形，並由原本一維延伸至二維之推廣。拼湊 L 形排法時有幾個要點，其一是 L 形的排法在排列時不可旋轉或翻轉，例如只取$\begin{matrix}- & \\ + & +\end{matrix}$爲一組，則$\begin{matrix}- & - & \\ + & + & +\end{matrix}$，$\begin{matrix}- & + & \\ + & + & -\end{matrix}$前者可被排成而後者則否，因爲排成後者時須要將排法翻轉；其二是除了邊界外每個位置上都有三個排法重疊，而在排法重疊的位置其+−號要相同，這也是從一維延伸至二維最大的困難。(如右圖)而我從研究的過程中發現一個解決相容性的方法，就是如果要排成 $m\times n$ 的圖形，可以先排成 $1\times n$ 或 $m\times 1$ 的排法，再以 $1\times n$ 或 $m\times 1$ 的排法爲基礎排出 $m\times n$ 的圖形。

觀察用矩陣計算直線形的過程，可以發現矩陣只記錄直線形中可以用來延伸的左右兩個邊界，因爲只有邊界上的+−號才會影響一個排法的衍伸排法。同樣的觀念推廣到平面上就是把一個排法四周可以再接下去的地方記錄在一個矩陣內。

因爲一個排法有上下左右四個邊界，所以必須把邊界的形態放在一個四維矩陣中(這裡的維度不是指向量空間的維度，而是把向量視爲一維，一般的矩陣視爲二維的維度。)對任何一個維度的第 n 列代表將 n 的 2 進位值 0→+,1→−後轉化成的圖樣，而矩陣中任何一個元爲邊界和所在行列所代表的圖樣相同的排法的個數。

爲了方便敘述，我將四維矩陣視爲 1 個大矩陣包著數個行列相同的小矩陣。其中大矩陣代表上下邊界；小矩陣代表左右邊界。所以 1 個四維矩陣 A 可表示如下：

$$A=\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix}=\begin{pmatrix} (a_{11ij})_{p\times q} & (a_{12ij})_{p\times q} & \cdots & (a_{1nij})_{p\times q} \\ (a_{21ij})_{p\times q} & (a_{22ij})_{p\times q} & & (a_{2nij})_{p\times q} \\ \vdots & & \ddots & \vdots \\ (a_{m1ij})_{p\times q} & (a_{m2ij})_{p\times q} & \cdots & (a_{mnij})_{p\times q} \end{pmatrix}=(a_{ijkl})_{m\times n\times p\times q}$$

下圖爲排成大小爲 2×2、2×3 時定義的四維矩陣

2×2	2×3
$\begin{matrix} & + & - \\ + & A_{11} & A_{12} \\ - & A_{21} & A_{22} \end{matrix}$	$\begin{matrix} & ++ & +- & -+ & -- \\ ++ & B_{11} & B_{12} & B_{13} & B_{14} \\ +- & B_{21} & B_{22} & B_{23} & B_{24} \\ -+ & B_{31} & B_{32} & B_{33} & B_{24} \\ -- & B_{41} & B_{42} & B_{43} & B_{44} \end{matrix}$

觀察後可發現，一個大小爲 $m\times n$ 的排法可用大小爲 $2^{n-1}\times 2^{n-1}\times 2^{m-1}\times 2^{m-1}$ 的四維矩陣，也就是一個大 $2^{n-1}\times 2^{n-1}$ 矩陣包著小 $2^{m-1}\times 2^{m-1}$ 矩陣。若定義運算 $P\otimes Q=\begin{pmatrix} p_{11}Q & p_{12}Q \\ p_{21}Q & p_{22}Q \end{pmatrix}=\begin{pmatrix} p_{11}q_{11} & p_{11}q_{12} & p_{12}q_{11} & p_{21}q_{12} \\ p_{11}q_{21} & p_{11}q_{22} & p_{21}q_{21} & p_{21}q_{22} \\ p_{21}q_{11} & p_{21}q_{12} & p_{22}q_{11} & p_{21}q_{12} \\ p_{21}q_{21} & p_{21}q_{22} & p_{21}q_{21} & p_{21}q_{22} \end{pmatrix}$,其中

$P=(p_{ij})_{2\times2}, Q=(q_{ij})_{2\times2}$，則 $B=A\otimes A$。

很自然的會想把上述性質推衍到一般的情況：

定義 3.1

$A\otimes B\equiv(c_{ij})_{m_1m_2\times n_1n_2}$ 稱為 A 增乘 B，其中

$c_{i_1(m_2-1)+i_2\ j_1(n_2-1)+j_2}=a_{i_1j_1}b_{i_2j_2},\ A=(a_{ij})_{m_1\times n_1},\ B=(b_{ij})_{m_2\times n_2}$。

定理 3.1

設Γ為一組 L 形的排法，A,B 分別代表Γ排成 $m\times n_1, m\times n_2$ 時的矩陣，則 $A\otimes B$ 代表Γ排成 $m\times(n_1+n_2-1)$時的矩陣。

證明

1.令 $C=A\otimes B$，其中 $A=(A_{ij})_{2^{n_1-1}\times2^{n_1-1}}, B=(B_{ij})_{2^{n_2-1}\times2^{n_2-1}}, C=(C_{ij})_{2^{(n_1+n_2-1)-1}\times2^{(n_1+n_2-1)-1}}$, $C_{i_1(2^{n_2-1}-1)+i_2\ j_1(2^{n_2-1}-1)+j_2}=A_{i_1j_1}B_{i_2j_2}$。對於任意由Γ排成大小為 $m\times n_1$ 的排法α，令β為一可和α直向相接大小為 $m\times n_2$ 的排法，γ為α,β相接成的排法。並將α,β,γ在 A,B,C 中的位置記為$(i_1,j_1,k_1,p),(i_2,j_2,p,l_2),(i_3,j_3,k_3,l_3)$。

2.因為γ為α,β直向相接成的排法，所以直向的邊界會接合而橫向的邊界會連長，因此 $k_3=k_1, l_3=l_2, i_3=i_1$ 的 2 進位值的每一位向左移 n_2 位$+i_2=i_1(2^{n_2-1}-1)+i_2$,同理 $j_3=j_1(2^{n_2-1}-1)+j_2$。

3.當 $C=A\otimes B$ 時，因為 $C_{i_1(2^{n_2-1}-1)+i_2\ j_1(2^{n_2-1}-1)+j_2}=A_{i_1j_1}B_{i_2j_2}$，依定義 $i_3=i_1(2^{n_2-1}-1)+i_2, j_3=j_1(2^{n_2-1}-1)+j_2$，又 $c_{i_1(2^{n_2-1}-1)+i_2\ j_1(2^{n_2-1}-1)+j_2\ k_3\ l_3}=\sum_{p=1}^{2^{n_2-1}}a_{i_1j_1k_1p}b_{i_2j_2pl_2}$，所以 $k_3=k_1, l_3=l_2$。

4.顯而易見，如果β和α不可直向相接，它們也代表的數也不會乘在一起，故本題得證。

上述都是朝左右延伸的情形，至於對上下延伸的情形，雖然不能直接用增乘，但可以間接的使用增乘得到：

定義 3.2

設 $A=(A_{ij})_{m\times n}, B=(B_{ij})_{n\times p}$，定義 $AB\equiv\left(\sum_{k=1}^{n}A_{ik}\otimes B_{kj}\right)_{m\times p}, A^n\equiv\overbrace{AA\cdots A}^{n個}$

定理 3.2

設Γ為一組 L 形的排法，A,B 分別代表Γ排成 $m_1\times n, m_2\times n$ 時的矩陣，則 AB 代表Γ排成$(m_1+m_2-1)\times n$ 時的矩陣。

其證明方式和定理 1 很類似，故從略。

以下以 $\begin{array}{ll}+ & \\ + & +\end{array}$, $\begin{array}{ll}+ & \\ - & -\end{array}$, $\begin{array}{ll}- & \\ + & -\end{array}$ 為例，說明實際的計算方式：

大小	排法	矩陣		個數
2×2	+ + + +'— —' — + —	A	$\begin{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&1\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\end{pmatrix}$	3
2×3	+ + + + +' + — + + —' + + — — —' — + + — —	$A\otimes A$	$\begin{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}&\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix}\\\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix}&\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix}&\begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix}\end{pmatrix}$ $=\begin{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&1\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\end{pmatrix}$	4

由前面的結論和「先排成 $1\times n$ 或 $m\times 1$ 的排法，再以 $1\times n$ 或 $m\times 1$ 的排法爲基礎排出 $m\times n$ 的圖形。」這個技巧可以得到下面定理。

定理 3.3

設Γ爲一組 L 形的排法，A 代表Γ排成 2×2 時的矩陣，B 代表Γ排成 $m\times n$ 時的矩陣，則 $B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\cdots A^{m-1}}^{n-1個}$

證明

1.當 $m=2,n=2$ 時 $B=A$，上式成立。

2.設 $m=2,n=\mathrm{k}$ 時 $B'=\overbrace{A\otimes A\otimes\cdots\cdots A}^{k-1個}$，則 $m=2,n=k+1$ 時

因爲Γ爲一組 L 形的排法，A,B' 分別代表Γ排成 $2\times2,2\times k$ 時的矩陣，所以 $A\otimes B'$ 代表Γ排成 $2\times(k+1)$時的矩陣，即

$$B=A\otimes B'=A\otimes\overbrace{A\otimes A\otimes\cdots\cdots A}^{k-1個}=\overbrace{A\otimes A\otimes\cdots\cdots A}^{k個}$$

3.由數學歸納法可知，當 $m=2$ 時 $B=\overbrace{A\otimes A\otimes\cdots\cdots A}^{n-1個}$ 成立。

4.同理可證 $B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\cdots A^{m-1}}^{n-1個}$ 成立。

(四) 加速矩陣計算的方法

不過這方法有個問題，因爲矩陣會逐漸變大，由小排法推出大排法的方法也會變得很複雜，所以我想了一些補救的方法。

因爲算直向時不用處理橫向的資料，所以我們可以將運算方式改爲用直線形的矩陣乘法計算，資料量就不會增加。

用數學式可表示爲

定義 4.1

設 $A=(A_{ijkl})_{m\times m\times n\times n}$，定義$\sum''A=(a_{00ij})_{n\times n},\sum'A=(a_{ij00})_{m\times m}$

$\sum'',\sum'$運算有以下性質：

$$(\sum''A)^n=\sum''(\overbrace{A\otimes A\otimes\cdots\cdots A}^{n個}),(\sum'A)^n=\sum'A^n$$

因爲兩式大同小異，故只證明左式

1.設 $A=\left(a_{ijkl}\right)_{m_1\times m_1\times n\times n},B=\left(b_{ijkl}\right)_{m_2\times m_2\times n\times n},f(x,y)=x-y\left[\frac{x}{y}\right]$，則

$$\sum''B\sum''A=(\mathrm{b}_{00ij})_{n\times n}(a_{00ij})_{n\times n}=\left(\sum_{k=1}^{n}b_{00ik}a_{00kj}\right)_{n\times n}$$

$$=\sum''\left(\left(\sum_{k=1}^{n}b_{\left\lfloor\frac{i}{m_2-1}\right\rfloor\left\lfloor\frac{j}{m_2-1}\right\rfloor ik}a_{f(i,m_2-1)\,f(i,m_2-1)\,kj}\right)_{n\times n}\right)_{m_1m_2\times m_1m_2}$$

$$=\sum''\left(B_{\left\lfloor\frac{i}{m_2-1}\right\rfloor\left\lfloor\frac{j}{m_2-1}\right\rfloor}A_{f(i,m_2-1)\,f(j,m_2-1)}\right)_{m_1m_2\times m_1m_2}=\sum''B\otimes A$$

2.當 $n=1$ 時，$\sum''A=\sum''A$ 顯然成立。

3.設 $n=k$ 時，$(\sum''A)^k=\sum''(\overbrace{A\otimes A\otimes\cdots\cdots A}^{k個})$成立，則 $n=k+1$ 時

$$(\sum''A)^{k+1}=(\sum''A)^k\sum''A=\sum''(\overbrace{A\otimes A\otimes\cdots\cdots A}^{k個})\sum''A=\sum''(\overbrace{A\otimes A\otimes\cdots\cdots A}^{k+1個})$$

4.由數學歸納法可知$(\sum''A)^n=\sum''(\overbrace{A\otimes A\otimes\cdots\cdots A}^{n個})$在 $n\in N$ 時成立。

可以發現，整個算法只有在做增乘時資料量會增加，所以要有效減少資料量只有在算時做增乘動手腳。

很多圖樣只會在排法的邊界出現，以致於矩陣空空洞洞的，所以如果把做增乘後的矩陣分成三個矩陣相乘，分別代表排法的上邊界、中間、下邊界，那就可以有效的減少矩陣的大小。

實際的計算方法是在做增乘後，刪除全爲 0 的行列，再分成三個矩陣，用數學式表示如下：

定義 4.2

設 $A=(a_{ij})_{m\times n},a_{ij}$ 是實數或矩陣，可令 $B=A+A^t$，則

$C=(c_{ij})_{l\times l}, c_{ij}=a_{p_i q_j}, b_{p_i 0}\neq 0, b_{0 q_j}\neq 0$ ，就是代表中間的矩陣；

$D=(d_{ij})_{m\times l}, d_{ij}=a_{i q_j}, b_{0 q_j}\neq 0$ ，是代表上邊界的矩陣；

$E=(e_{ij})_{l\times n}, e_{ij}=a_{p_i j}, b_{p_i 0}\neq 0$ ，是代表下邊界的矩陣；其中 $p_i<p_{i+1}, q_j<q_{j+1}$,而 l 由實際情況決定，計算前無法預料。定義 md $A=C$,ld $A=D$,rd $A=E$。

由此定義可述定理 4.1：

定理 4.1

若 $A=(a_{ij})_{n\times n}, a_{ij}$ 是實數或矩陣，$A^g=\text{ld } A\,(\text{md } A)^{g-2}\,\text{rd } A$，其中 $g\in N-\{1\}$

例如：

本例所用的排法和上節相同。

		矩陣
第一步	A	$\begin{pmatrix} \begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \end{pmatrix}$
第二步	ld A md A rd A	$\begin{pmatrix} \begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \end{pmatrix} \begin{pmatrix} \begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \end{pmatrix} \begin{pmatrix} \begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \\ \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix} \end{pmatrix}$

故可得知由最初的矩陣推衍的方法，如欲增乘已分割的矩陣，只要分別對它們做⊗運算就好了。

結論

矩陣的解題步驟

(一)先求出代表排法Γ的矩陣 A。

(二)由定理 3 可知，$T(\Gamma,m,n)=B$ 中元的和，其中 $B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\cdots A^{m-1}}^{n-1\text{個}}$

(三)由性質 8 和定理 4 可知，令 $C=\text{ld } A, D=\text{md } A, E=\text{rd } A$，則

$$B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\cdots A^{m-1}}^{n-1\text{個}}=\overbrace{CD^{m-1}E\otimes CD^{m-1}E\cdots\otimes CD^{m-1}E}^{n-1\text{個}}$$

$$=\overbrace{C\otimes C\cdots\otimes C}^{n-1\text{個}}\times\overbrace{D\otimes D\cdots\otimes D}^{n-1\text{個}}{}^{m-1}\times\overbrace{E\otimes E\cdots\otimes E}^{n-1\text{個}}$$

(四)綜合上述步驟可得

$$T(\Gamma,m,n)=\sum{}'B=\sum{}'\overbrace{C\otimes C\cdots\otimes C}^{n-1個}\times\left(\sum{}'\overbrace{D\otimes D\cdots\otimes D}^{n-1個}\right)^{m-1}\times\sum{}'\overbrace{E\otimes E\cdots\otimes E}^{n-1個}$$

矩陣的解題步驟中 1.把排法化成矩陣,2.做矩陣乘法,4.加起來 是基本的解法，而 3.分解成三個矩陣主要用來加速運算。就加速的效果來看，分解成三個矩陣的適用範圍較小，而加速效果較大。

討論

目前理論上可以求出任意大小時的排法個數，實際上在不加速的情況下要記錄大小 $m\times n$ 的排法就要 2^{mn} bits 的資料量，以我家的硬碟 $\log_2\dfrac{14323MB}{1bit}\approx 36$ 最多可以算到 6×6，所以非常不實用；加上第一個加速方法計算大小 $m\times n$ 的排法要 $2^{min(2m,2n)}$ bits，就好多了；第二個加速方法並不適合這種計算，條件適當時用手算就可以算出任意 $m\times n$ 時的情形，反之則沒什麼幫助。

一維時就像馬可夫鏈只要求出轉移矩陣，對角化後就可以輕易知道任意長度時的個數。針對二維的情形，目前已有一定步驟的解題方法，希望對增乘有更深入的研究後，可以達到像馬可夫鏈一樣的便捷。

參考書目

(一)林松山，細胞神經網路模型的數學研究，科學發展月刊 第 28 卷 第 3 期。

(二)班榮超，一維細胞神經網絡之穩定解個數研究(Number of Stable Equilibria in One Dimensional Cellular Neural Network)，國立交通大學應用數學系碩士論文。

(三)賴漢卿著，簡易線性代數，大中國圖書公司。

(四)游士賢著，線性代數，立功出版社。

(五)Microsoft®公司，MSDN Library Visual Studio 6.0 版。

Number of Stable Equilibria in Two Dimensional Cellular Neural Network

1. Motivation

J. Ban [1] studied the Number of Stable Equilibria in One Dimensional Cellular Neural Network by the patterns of size n; $n \geq 3$ is odd. After discussing with my teacher, I extend his result to two-dimensional dented rectangular patterns constructed from a given subclass Γ of L-shaped template and give an algorithm to compute them.

2. Purpose

(A) To discuss and investigate the patterns of two-dimensional Cellular Neural Network (CNN).

(B) To compute the number of the admissible dented rectangular patterns by using matrix multiplication.

3. Procedures

(A) Notation and definition

In this paper, capital English letters A, B, C, D, E and M represent matrices;

small English italic letters a, b, c, g, i, j, k, l, m, n, p, q, r and s represent integers;

a capital Greek letter Γ represents a set of templates and Φ and Θ represents a set of same size or same shape patterns;

small Greek letter α, β, γ, δ, ε represent signed patterns;

small English normal letters x, y represent sign, + or −.

a_{ij} represents the value of (i, j) entry of the $m \times n$ matrix A and $a_{*j} \equiv \sum_{i=1}^{m} a_{ij}$, $a_{i*} \equiv \sum_{j=1}^{n} a_{ij}$, $a_{**} \equiv \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}$, $\mathbf{0}$ denotes the zero matrix; $M(\alpha)$ is a matrix that records pattern α; $t(\Gamma, m\times n)$ represent the number of $m\times n$ dented rectangular patterns constructed from Γ; $\Theta(\Gamma, m\times n)$ represent all admissible $m\times n$ dented rectangular patterns with respect to Γ.

(B) Background

A linear pattern of size n is a $1\times n$ tabular with + or − signs on each square. In particular a pattern of size 3 is called a linear template. Hence, there are 8 different signed linear templates, (+++),(++−),(+−+), (+−−),(−++),(−+−),(−−+),(−−−). Two linear templates α and β can be concatenated if the last 2 signs in α are the same as the first 2 signs in β, i.e. $\alpha = \alpha_1$xy, β = xyβ_1, where x, y$\in\{+,-\}$ and α_1 and β_1 are sub-linear patterns of α and β, respectively, so we denote that $\alpha \oplus \beta = \alpha_1xy\beta_1$. Given a subclass Γ of the set of 8 different linear templates, we would like to construct a linear pattern of size n by using elements in Γ with operator $\oplus$. For example, let Γ = {α = ++−, β = +−+, γ = +−−, δ = −+−}, then we can construct 4 different patterns of size 5. They are

++−+− = $\alpha \oplus \beta \oplus \delta$, +−+−+ = $\beta \oplus \delta \oplus \beta$, +−+−− = $\beta \oplus \delta \oplus \gamma$, −+−+− = $\delta \oplus \beta \oplus \delta$.

These 4 different linear patterns are admissible with respect to Γ, but +++++ cannot be constructed by using elements in Γ. Hence, we say that +++++ is not admissible with respect to Γ.

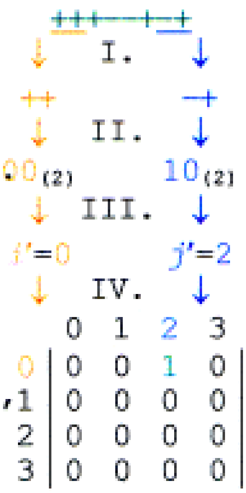
The method [1] is to record patterns α into a matrix $M(\alpha)$. Let's illustrate how to do this by an example:

I. Pick out the first 2 signs and the last 2 signs in α, which are ++,−+ respectively in this example.

II. Replace + with 0, − with 1 to change the first and last 2 signs into two-digit numbers, respectively.

III. We can view these two-digit numbers as binary numbers, and call them i', j', respectively.

IV. Construct the 4×4 matrix $M(\alpha)$ by $m_{ij}=1$ if $i=i'$, $j=j'$; otherwise, $m_{ij}=0$.



If we want to record a set Φ of patterns into a matrix $M(\Phi)$, we first construct the matrix $M(\alpha)$ recording for each pattern in this set Φ, and let $M(\Phi)=\sum_{\alpha\in\Phi} M(\alpha)$.

The advantage of this way of recording is: $M(\alpha)M(\beta)=M(\alpha\oplus\beta)$. For example:

α = +++−−+−+, β = −++++−+−, $\alpha\oplus\beta$ = +++−−+−++++−+−

$$M(\alpha)=\begin{pmatrix}0&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}, M(\beta)=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&1&0&0\\0&0&0&0\end{pmatrix}, M(\alpha\oplus\beta)=M(\alpha)M(\beta)=\begin{pmatrix}0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}.$$

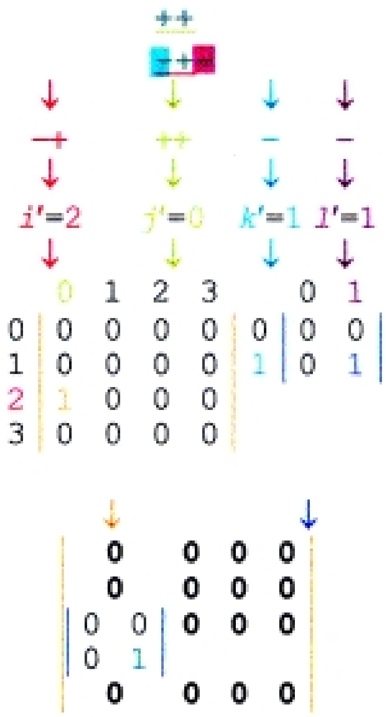
For any given set Γ of linear templates, we can record this set $\Theta(\Gamma, n)$ into a matrix $M(\Theta(\Gamma, n))$. It was proved that $M(\Theta(\Gamma, n))=M(\Gamma)^{n-2}$ in [1]. So the total number $t(\Gamma,n)$ of admissible patterns of size n with respect to Γ is exactly the sum of all entries in $M(\Gamma)^{n-2}$.

(C) 2-dimensional patterns

An $m\times n$ dented rectangular pattern is an $m\times n$ tabular with the upper right square missing, in particular, a 2×2 dented rectangle is called L-shaped template. We would like to label each square in an $m\times n$ dented rectangular pattern with + or − signs using, iteratively, a element chosen from a given subclass Γ of the eight different signed L-shaped templates: $\left(\begin{smallmatrix}+&\\+&+\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}+&\\+&-\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}+&\\-&+\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}+&\\-&-\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}-&\\+&+\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}-&\\+&-\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}-&\\-&+\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}-&\\-&-\end{smallmatrix}\right)$. When two L-shaped templates are overlapped, it is required that the same sign on the overlapped square. A signed dented rectangle is called admissible patterns with respect to Γ if it can be constructed in this way.

There are two ways to construct a pattern from two L-shaped template α and β. That let $\alpha = \begin{smallmatrix}x&\\y&z\end{smallmatrix}$, $\beta = \begin{smallmatrix}y&\\p&q\end{smallmatrix}$, $\gamma = \begin{smallmatrix}s&\\z&t\end{smallmatrix}$, then we define $\delta = \alpha \oplus_v \beta = \begin{smallmatrix}x&\\y&z\\p&q\end{smallmatrix}$, $\varepsilon = \alpha \oplus_h \gamma = \begin{smallmatrix}x&s&\\y&z&t\end{smallmatrix}$, and we call δ is constructed from α and β vertically and ε is constructed from α and γ horizontally. For example, let $\Gamma=\{\alpha=\genfrac{}{}{0pt}{}{+}{++},\ \beta=\genfrac{}{}{0pt}{}{+}{-+},\ \gamma=\genfrac{}{}{0pt}{}{-}{++}\}$. Then there are 6 different of 2×3 admissible patterns with respect to Γ; 6 different of 3×2 admissible patterns constructed with respect to Γ: $\genfrac{}{}{0pt}{}{++}{+++}=\alpha\oplus_h\alpha$, $\genfrac{}{}{0pt}{}{++}{+-+}=\alpha\oplus_h\gamma$, $\genfrac{}{}{0pt}{}{++}{-++}=\beta\oplus_h\alpha$, $\genfrac{}{}{0pt}{}{+-}{-++}=\beta\oplus_h\gamma$, $\genfrac{}{}{0pt}{}{-+}{+++}=\gamma\oplus_h\alpha$, $\genfrac{}{}{0pt}{}{--}{+++}=\gamma\oplus_h\gamma$, 5 different of 3×2 admissible patterns constructed with respect to Γ: $\begin{smallmatrix}+\\++\\++\end{smallmatrix}=\alpha\oplus_v\alpha$, $\begin{smallmatrix}-\\++\\++\end{smallmatrix}=\alpha\oplus_v\gamma$, $\begin{smallmatrix}+\\++\\-+\end{smallmatrix}=\beta\oplus_v\alpha$, $\begin{smallmatrix}-\\++\\-+\end{smallmatrix}=\beta\oplus_v\gamma$, $\begin{smallmatrix}+\\-+\\++\end{smallmatrix}=\gamma\oplus_v\beta$.

As we've mentioned in the one-dimensional case, the main spirit of using matrices for recording and computing patterns is to record only the boundary part, because it is precisely the boundary part of a pattern that determines which and how many new patterns that can be constructed. Now it is the same in two-dimensional case. So the first step is to find a way to record the boundary of a two-dimensional pattern into a matrix.



Given any $m\times n$ pattern α, because it has four sides we have to record the signs on the boundary of α into a four index matrix $(a_{ijkl})_{p\times q\times r\times s}$, where $p=q=2^{n-1}$, $r=s=2^{m-1}$.

For example, let $\alpha=\genfrac{}{}{0pt}{}{++}{-+-}$.

I. Pick out the bottom linear pattern (−+) of size n−1 (ignore the last sign). Let i' be the decimal number represents this linear pattern just like we have done in Section (2). Similarly, j' is the decimal number represents the top linear pattern (++). In this example, $i'=2$ and $j'=0$.

II. Pick out the left boundary as a linear pattern (−) of size m−1. Note that we read it from the bottom to the top, and we ignore the top sign. Let k' be the decimal number represents this linear pattern. Similarly, l' is the decimal number represents the right boundary pattern (−). In this example, $k'=1$ and $l'=1$.

III. So the matrix $M(\alpha)$ representing pattern α has $m_{i'j'k'l'}=1$ and 0 for other entries.

For convenience, I view a "4-dimensional" matrix as a matrix whose every entry is itself a matrix. Hence a "4-dimensional" matrix A can be expressed as follows:

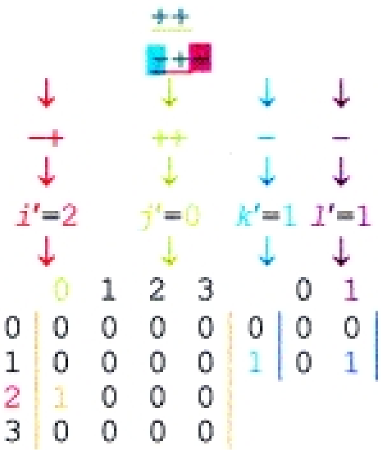
$$A=\begin{pmatrix}A_{00} & A_{01} & \cdots & A_{0(n-1)}\\ A_{01} & A_{11} & & A_{0(n-1)}\\ \vdots & & \ddots & \vdots\\ A_{(m-1)0} & A_{(m-1)1} & \cdots & A_{(m-1)(n-1)}\end{pmatrix}=\begin{pmatrix}(a_{00kl})_{p\times q} & (a_{01kl})_{p\times q} & \cdots & (a_{0(n-1)kl})_{p\times q}\\ (a_{10kl})_{p\times q} & (a_{11kl})_{p\times q} & & (a_{1(n-1)kl})_{p\times q}\\ \vdots & & \ddots & \vdots\\ (a_{(m-1)0kl})_{p\times q} & (a_{(m-1)1kl})_{p\times q} & \cdots & (a_{(m-1)(n-1)kl})_{p\times q}\end{pmatrix}$$

Just like the one dimension case, we wonder if this matrix representation can be used to construct bigger patterns by certally. For example, let $\Gamma=\{\alpha=\overset{+}{++},\ \beta=\overset{+}{-+},\ \gamma=\overset{-}{++}\}$. Then there are 6 different of 2×3 admissible patterns with respect to Γ; 6 different of 3×2 admissible patterns constructed with respect to Γ: $\overset{++}{+++}=\alpha\oplus_h\alpha$, $\overset{++}{+-+}=\alpha\oplus_h\gamma$, $\overset{++}{-++}=\beta\oplus_h\alpha$, $\overset{+-}{-++}=\beta\oplus_h\gamma$, $\overset{-+}{+++}=\gamma\oplus_h\alpha$, $\overset{--}{+++}=\gamma\oplus_h\gamma$, 5 different of 3×2 admissible patterns constructed with respect to Γ: $\begin{smallmatrix}+\\++\\++\end{smallmatrix}=\alpha\oplus_v\alpha$, $\begin{smallmatrix}-\\++\\++\end{smallmatrix}=\alpha\oplus_v\gamma$, $\begin{smallmatrix}+\\++\\-+\end{smallmatrix}=\beta\oplus_v\alpha$, $\begin{smallmatrix}-\\++\\-+\end{smallmatrix}=\beta\oplus_v\gamma$, $\begin{smallmatrix}+\\-+\\++\end{smallmatrix}=\gamma\oplus_v\beta$.

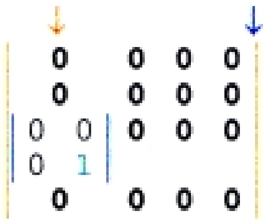
As we've mentioned in the one-dimensional case, the main spirit of using matrices for recording and computing patterns is to record only the boundary part, because it is precisely the boundary part of a pattern that determines which and how many new patterns that can be constructed. Now it is the same in two-dimensional case. So the first step is to find a way to record the boundary of a two-dimensional pattern into a matrix.

Given any $m\times n$ pattern α, because it has four sides we have to record the signs on the boundary of α into a four index matrix $(a_{ijkl})_{p\times q\times r\times s}$, where $p=q=2^{n-1}$, $r=s=2^{m-1}$.

For example, let $\alpha=\overset{++}{-+-}$.



I. Pick out the bottom linear pattern (-+) of size n-1 (ignore the last sign). Let i' be the decimal number represents this linear pattern just like we have done in Section (2). Similarly, j' is the decimal number represents the top linear pattern (++). In this example, $i'=2$ and $j'=0$.



II. Pick out the left boundary as a linear pattern (-) of size m-1. Note that we read it from the bottom to the top, and we ignore the top sign. Let k' be the decimal number represents this linear pattern. Similarly, l' is the decimal number represents the right boundary pattern (-). In this example, $k'=1$ and $l'=1$.

III. So the matrix $M(\alpha)$ representing pattern α has $m_{i'j'k'l'}=1$ and 0 for other entries.

For convenience, I view a "4-dimensional" matrix as a matrix whose every entry is itself a matrix. Hence a "4-dimensional" matrix A can be expressed as follows:

$$A=\begin{pmatrix}A_{00} & A_{01} & \cdots & A_{0(n-1)}\\ A_{01} & A_{11} & & A_{0(n-1)}\\ \vdots & & \ddots & \vdots\\ A_{(m-1)0} & A_{(m-1)1} & \cdots & A_{(m-1)(n-1)}\end{pmatrix}=\begin{pmatrix}(a_{00kl})_{p\times q} & (a_{01kl})_{p\times q} & \cdots & (a_{0(n-1)kl})_{p\times q}\\ (a_{10kl})_{p\times q} & (a_{11kl})_{p\times q} & & (a_{1(n-1)kl})_{p\times q}\\ \vdots & & \ddots & \vdots\\ (a_{(m-1)0kl})_{p\times q} & (a_{(m-1)1kl})_{p\times q} & \cdots & (a_{(m-1)(n-1)kl})_{p\times q}\end{pmatrix}$$

Just like the one dimension case, we wonder if this matrix representation can be used to construct bigger patterns by certain operation. The following is the example to construct 2×3 patterns from 2×2 templates in view of matrices.

2×2	2×3
$\begin{pmatrix}\begin{pmatrix}a_0&a_1\\0&0\end{pmatrix}&\begin{pmatrix}a_4&a_5\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\a_2&a_3\end{pmatrix}&\begin{pmatrix}0&0\\a_6&a_7\end{pmatrix}\end{pmatrix}$	$\begin{pmatrix}\begin{pmatrix}a_0^2&a_0a_1\\0&0\end{pmatrix}&\begin{pmatrix}a_0a_4&a_0a_5\\0&0\end{pmatrix}&\begin{pmatrix}a_4a_0&a_4a_1\\0&0\end{pmatrix}&\begin{pmatrix}a_4^2&a_4a_5\\0&0\end{pmatrix}\\\begin{pmatrix}a_1a_2&a_1a_3\\0&0\end{pmatrix}&\begin{pmatrix}a_1a_6&a_1a_7\\0&0\end{pmatrix}&\begin{pmatrix}a_5a_2&a_5a_3\\0&0\end{pmatrix}&\begin{pmatrix}a_5a_6&a_5a_7\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\a_2a_0&a_2a_1\end{pmatrix}&\begin{pmatrix}0&0\\a_2a_4&a_2a_5\end{pmatrix}&\begin{pmatrix}0&0\\a_6a_0&a_6a_1\end{pmatrix}&\begin{pmatrix}0&0\\a_6a_4&a_6a_5\end{pmatrix}\\\begin{pmatrix}0&0\\a_3a_2&a_2^2\end{pmatrix}&\begin{pmatrix}0&0\\a_3a_6&a_3a_7\end{pmatrix}&\begin{pmatrix}0&0\\a_7a_2&a_7a_3\end{pmatrix}&\begin{pmatrix}0&0\\a_7a_6&a_7^2\end{pmatrix}\end{pmatrix}$

where a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7 are the number of $\left(\begin{smallmatrix}+&\\+&+\end{smallmatrix}\right)$,$\left(\begin{smallmatrix}+&\\+&-\end{smallmatrix}\right)$,$\left(\begin{smallmatrix}+&\\-&+\end{smallmatrix}\right)$,$\left(\begin{smallmatrix}+&\\-&-\end{smallmatrix}\right)$,$\left(\begin{smallmatrix}-&\\+&+\end{smallmatrix}\right)$,$\left(\begin{smallmatrix}-&\\+&-\end{smallmatrix}\right)$,$\left(\begin{smallmatrix}-&\\-&+\end{smallmatrix}\right)$,$\left(\begin{smallmatrix}-&\\-&-\end{smallmatrix}\right)$ in a given set Γ, i.e. a_i=1, if the ith L-shape pattern belongs to Γ; otherwise, a_i=0. this example suggests an operation:

$$D\otimes E=\begin{pmatrix}D_{00}E & D_{01}E\\ D_{10}E & D_{11}E\end{pmatrix}=\left(\begin{array}{cc|cc}D_{00}E_{00} & D_{00}E_{01} & D_{01}E_{00} & D_{01}E_{01}\\ D_{00}E_{10} & D_{00}E_{11} & D_{01}E_{10} & D_{01}E_{11}\\ \hline D_{10}E_{00} & D_{10}E_{01} & D_{22}E_{00} & D_{22}E_{01}\\ D_{10}E_{10} & D_{10}E_{11} & D_{22}E_{10} & D_{22}E_{11}\end{array}\right),$$

where $D=(d_{ij})_{2\times2\times2\times2}$, $E=(e_{ij})_{2\times2\times2\times2}$

We observe that a $m\times n$ pattern can be represented as a 4-index matrix of size $2^{n-1}\times2^{n-1}\times2^{m-1}\times2^{m-1}$, i.e. a $2^{n-1}\times2^{n-1}$ matrix whose every entry is a $2^{m-1}\times2^{m-1}$ matrix. In general, we define an operation of matrices for the horizontal construction as follows:

Definition 3.1

Let $A=(A_{ij})_{m_1\times n_1}, B=(B_{ij})_{m_2\times n_2}$ be four-index matrices with A_{ij}, and B_{ij} being real numbers or matrices such that the multiplication of matrices $A_{i_1j_1}B_{i_2j_2}$ is well defined. We define

$$A\otimes B=(C_{ij})_{m_1m_2\times n_1n_2} \text{ with } C_{i_1m_2+i_2,\ j_1n_2+j_2}=A_{i_1j_1}B_{i_2j_2}.$$

Theorem 3.1

Let Γ be a set of L-shaped templates. Suppose A and B, respectively, are the matrices that represent all admissible $m\times n_1$ and $m\times n_2$ patterns with respect to Γ. Then $A\otimes B$ is the matrix representing all admissible $m\times(n_1+n_2-1)$ patterns with respect to Γ.

Proof

I. Let α and β, respectively, be $m\times n_1$ and $m\times n_2$ patterns that can be constructed with respect to Γ. Suppose an $m\times(n_1+n_2-1)$ pattern γ is constructed by α and β horizontally, and α, β and γ are recorded in the (i_1,j_1,k_1,p), (i_2,j_2,p,l_2) and (i_3,j_3,k_3,l_3) entries in the matrices $A=M(\alpha)$, $B=M(\beta)$ and $C=M(\gamma)$, respectively.

II. Because the pattern γ is constructed from α and β horizontally, the right side of α and the left side of β glue together, and the top (bottom) sides of α and β are concatenated. So $k_3=k_1$, $l_3=l_2$, i_3=(shifting every digit in the binary expression of i_1 to the left by n_2 digits)+i_2 = $i_1 2^{n_2-1}+i_2$. Similarly, $j_3=j_1 2^{n_2-1}+j_2$.

III. The argument implies that $a_{i_1 j_1 k_1 p}$ and $b_{i_2 j_2 p l_2}$ contribute $\left(i_1 2^{n_2-1}+i_2,\ j_1 2^{n_2-1}+j_2,\ k_1,\ l_2\right)$ entry of C with $a_{i_1 j_1 k_1 p} b_{i_2 j_2 p l_2}$. Therefore

$$c_{i_1 2^{n_2-1}+i_2\ j_1 2^{n_2-1}+j_2\ k_1\ l_2} = \sum_p a_{i_1 j_1 k_1 p} b_{i_2 j_2 p l_2} = \left(A_{i_1 j_1} B_{i_2 j_2}\right)_{k_1 l_2} \text{ and}$$

$$C_{i_1 2^{n_2-1}+i_2\ j_1 2^{n_2-1}+j_2} = A_{i_1 j_1} B_{i_2 j_2}$$

The discussion above is to construct patterns horizontally. Although we don't directly use $\otimes$ operation to construct patterns vertically, we can use $\otimes$ operation indirectly:

Definition 3.2

For two-index matrices A and B, we set AB be the ordinary matrix multiplication. For four-index matrices A and B where $A=(A_{ij})_{m\times n}$ and $B=(B_{ij})_{n\times p}$, we set $AB = \left(\sum_{k=1}^{n} A_{ik} \otimes B_{kj}\right)_{m\times p}$. Also we define

$$A^n = \overbrace{AA\cdots A}^{n}.$$

Theorem 3.2

Let Γ be a set of L-shaped templates. Suppose A and B, respectively, are the matrices that represent all admissible $m_1\times n$ and $m_2\times n$ patterns with respect to Γ. Then AB is the matrix representing all admissible $(m_1+m_2-1)\times n$ patterns with respect to Γ.

The proof is similar to theorem 3.1, so we omit it.

Below we use $\Gamma=\{\begin{smallmatrix}+ & \\ + & +\end{smallmatrix}, \begin{smallmatrix}+ & \\ - & -\end{smallmatrix}, \begin{smallmatrix}- & \\ + & -\end{smallmatrix}\}$ as an example to illustrate how to actually carry out the computation:

Size	Pattern	Matrix		Number
2×2	+ + + +, − − − , + −	A	$\begin{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix}\\ \begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix}\end{pmatrix}$	3
2×3	+ + + + +, + − + + −, + + − − −, − + + − −	$A\otimes A$	$\begin{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix}\\ \begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix}\\ \begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}0&1\\0&0\end{pmatrix}\\ \begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}0&0\\0&0\end{pmatrix}\begin{pmatrix}0&0\\0&0\end{pmatrix}\end{pmatrix}$	4

			$=\begin{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&1\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&1\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}&\begin{pmatrix}0&0\\0&0\end{pmatrix}\end{pmatrix}$	
3×2	+ + +, + + − + −, + + + − −, + − − + −, − − + + + − −	A^2	$\begin{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}\otimes\begin{pmatrix}1&0\\0&0\end{pmatrix}+\begin{pmatrix}0&1\\0&0\end{pmatrix}\otimes\begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}1&0\\0&0\end{pmatrix}\otimes\begin{pmatrix}0&1\\0&0\end{pmatrix}+\begin{pmatrix}0&1\\0&0\end{pmatrix}\otimes\begin{pmatrix}0&0\\0&0\end{pmatrix}\\\begin{pmatrix}0&0\\0&1\end{pmatrix}\otimes\begin{pmatrix}1&0\\0&0\end{pmatrix}+\begin{pmatrix}0&0\\0&0\end{pmatrix}\otimes\begin{pmatrix}0&0\\0&1\end{pmatrix} & \begin{pmatrix}0&0\\0&1\end{pmatrix}\otimes\begin{pmatrix}0&1\\0&0\end{pmatrix}+\begin{pmatrix}0&0\\0&0\end{pmatrix}\otimes\begin{pmatrix}0&0\\0&0\end{pmatrix}\end{pmatrix}$ $=\begin{pmatrix}\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{pmatrix} & \begin{pmatrix}0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\\\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\end{pmatrix}$ $=\begin{pmatrix}\begin{pmatrix}1&0&0&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{pmatrix}&\begin{pmatrix}0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\\\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&0\end{pmatrix}&\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{pmatrix}\end{pmatrix}$	5

Summarizing the discussion above, we get the following theorem. The first equality represents constructing vertically first and then horizontally; the last equality represents constructing horizontally first and then vertically.

Theorem 3.3

Let Γ be a set of L-shaped templates, $A=M(\Gamma)$ and $B=M(\Theta(\Gamma, m\times n))$. Then $B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\otimes A^{m-1}}^{n-1}=\left(\overbrace{A\otimes A\otimes\cdots\otimes A}^{n-1}\right)^{m-1}$

We prove the first equality only, and the second one is similar.

Proof

I. When $m=2$, $n=2$, then $B=A$.

II. If $m=2$, $n=k$, $B'=\overbrace{A\otimes A\otimes\cdots\otimes A}^{k-1}$, then when $m=2$, $n=k+1$, A, B' represent 2×2, $2\times k$ patterns constructed from Γ, respectively, so $A\otimes B'$ represent $2\times(k+1)$ patterns constructed from Γ, i.e.

$B=A\otimes B'=A\otimes\overbrace{A\otimes A\otimes\cdots\otimes A}^{k-1}=\overbrace{A\otimes A\otimes\cdots\otimes A}^{k}$

III. Hence by induction, when m=2, we have $B=\overbrace{A\otimes A\otimes\cdots\otimes A}^{n-1}$.

IV. Similarly, for any positive integer m (≥2), we have

$B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\otimes A^{m-1}}^{n-1}$.

(D) The first way to accelerate computation

Our method has a serious drawback: the bigger dented rectangular pattern, the larger the matrix. The computation involved will become very complex when we construct a large pattern, so we thought of some ways to solve this problem.

We will process the horizontal data after we construct all $m\times2$ patterns vertically, so we can modify the 4-index matrix and ⊗ operation into 2-index matrix and the ordinary matrix multiplication, then the data volume won't increase.

We can formulate this idea as follows.

Definition 4.1

Define $\sum' A, \sum'' A$ to be $(a_{ij**})_{m\times m}$, $(a_{**kl})_{n\times n}$, respectively, where $A=(a_{ijkl})_{m\times m\times n\times n}$.

$\sum'$, $\sum''$ operation have following properties :

Property $(\sum'' A)^n=\sum''(\overbrace{A\otimes A\otimes\cdots\otimes A}^{n})$, $(\sum' A)^n=\sum' A^n$

Because the two equalities are similar, we prove the first one.

Proof

I. $$\sum'' A\otimes B=\sum_{i,j,k,l}A_{ij}B_{kl}=\sum_{i,j}A_{ij}\sum_{k,l}B_{kl}=\sum'' A\sum'' B$$

II. When n=1, we have $\sum'' A=\sum'' A$.

III. If $n=k$, $(\sum'' A)^k=\sum''(\overbrace{A\otimes A\otimes\cdots\otimes A}^{k})$ hold, then when $n=k+1$,

$$(\sum'' A)^{k+1}=(\sum'' A)^k\sum'' A=\sum''(\overbrace{A\otimes A\otimes\cdots\otimes A}^{k})\sum'' A=\sum''(\overbrace{A\otimes A\otimes\cdots\otimes A}^{k+1})$$

IV. Hence by induction, we have $(\sum'' A)^n=\sum''(\overbrace{A\otimes A\otimes\cdots\otimes A}^{n})$ when $n\in\mathbf{N}$.

(E) The second way to accelerate computation

We discover that the volume of data will increase only when we use ⊗ operation in this scheme, so if we want to reduce the volume of data effectively, we should do some tricks when we use ⊗ operation.

We find that the matrix representing a set of patterns has many zero entries, so if we divide the matrix representing a set of patterns into three matrices that represent the patterns top boundary, middle part and bottom boundary, respectively. Then we can reduce the size of the matrix effectively.

The actual computation method is to erase the columns and rows with all zero entries after using ⊗ operation, and then divide it into three matrices.

4. Conclusion

We summarize the scheme to count the patterns following these procedures:

(A) Construct the matrix A representing Γ.

(B) According to theorem 3.3, $t(\Gamma, m\times n)=b_{\ast\ast\ast\ast}$, where

$$B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\otimes A^{m-1}}^{n-1}$$

(C) According to theorem 5.2, let $C=t(A)$, $D=m(A)$, $E=b(A)$, then

$$B=\overbrace{A^{m-1}\otimes A^{m-1}\otimes\cdots\otimes A^{m-1}}^{n-1}=\overbrace{CD^{m-3}E\otimes CD^{m-3}E\otimes\cdots\otimes CD^{m-3}E}^{n-1}$$

$$=\overbrace{C\otimes C\otimes\cdots\otimes C}^{n-1}\overbrace{D\otimes D\otimes\cdots\otimes D}^{n-1}{}^{m-3}\ \overbrace{E\otimes E\otimes\cdots\otimes E}^{n-1}$$

Step 1 in this program is to construct the matrix representing patterns; step 2 is to multiply matrices; step 4 is the main point in this program, and step 3 is for speeding up the computation. The method of dividing the matrix has a greater speeding effect, but it is not always applicable.

5. Discussion

In theory, we can compute the number of patterns no matter how big it is. In practice, we have to use 2^{mn} bits for recording $m\times n$ dented rectangular pattern if the speeding methods are not applied. The volume of my hard disk, $(\log_2 \frac{14323MB}{1bit} \approx 36)$ can only compute up to 6x6,so it is not very practical, we only have to use $2^{\min(2m,2n)}$ bits for recording with the first speeding method, this is much better. The effect of the second speeding method depends case by case; it won't always help.

In the one-dimensional case the situation is the same as the Markov chain all we have to do is diagonalize the transition matrix, then we can easily compute the number of patterns of any size. For the two-dimensional case, since now we've come up with an algorithm, we hope after gaining more understanding of the $\otimes$ operation we can make the computation as easy as that of the Markov chain.

6. References

(A) Lin Song-Sun, mathematical theory of Cellular neural network, National science council monthly V. 28 N. 3. (1999)

(B) Ban Jung-Chao, Number of Stable Equilibria in One Dimensional Cellular Neural Network. (1999)